Università degli Studi di Bologna
Scuola di Ingegneria

Corso di
Reti di Calcolatori T

Esercitazione 3 (proposta)
Selezione Parola / Elimina Linea
Socket C senza e con connessione

Antonio Corradi, Luca Foschini
Lorenzo Rosa, Giuseppe Martuscelli, Marco Torello
Anno Accademico 2022/2023

SOCKET SENZA CONNESSIONE – CLIENT PAROLA PIÙ LUNGA

Sviluppare un'applicazione C/S che realizzi un **servizio che fornisce informazioni sulla parola più lunga** in un file di testo presente su un file server remoto

Una parola è definita come una **sequenza di caratteri di lunghezza qualunque e delimitata da separatori** (consideriamo solo *inizio linea* / *fine linea e spazi*; pensare anche a come fare per averne altri …)

Il **Client** è un filtro ben fatto: chiede ciclicamente all'utente **il nome di un file remoto**, quindi invia al server una **richiesta con il nome del file** e attende dal server la risposta che è che indica **il numero di caratteri della parola più lunga del file**
il risultato è **un intero**, se il file esiste sul server, altrimenti **una notifica di errore**, nel caso il file remoto non esista (o sia vuoto o…altri errori)

In ogni caso il cliente stampa a video la risposta

SOCKET SENZA CONNESSIONE – SERVER PAROLA PIÙ LUNGA

Il **Server** **attende richieste** dai clienti: riceve il nome del file quindi, se il file esiste presso di lui, lo analizza per identificare la parola formata dal maggior numero di lettere e dare una risposta al cliente

Effettuata la identificazione della parola, **invia al client un intero** che indica il **numero di lettere** che formano la parola più lunga del file testo; altrimenti invia al client **un messaggio di errore** (ad esempio -1)

Il server può essere realizzato come **server**
sequenziale o
concorrente e parallelo

SOCKET CON CONNESSIONE: SPECIFICA ELIMINA LINEA DA FILE

Sviluppare un'applicazione C/S per **l'eliminazione di una linea** all'interno **di un file di testo presente** sul file system del cliente stesso

L'operazione prevede:
che l'utente fornisca **il nome del file** e **il numero di linea,**
che il cliente fornisca **il numero di linea e il contenuto di un file** al server,
che il server riceva il file ed **elimini la linea**, **spedendo il nuovo contenuto** (senza la linea specificata dal cliente) al client

SOCKET CON CONNESSIONE: CLIENT/SERVER ELIMINA LINEA DA FILE

Il **Client** chiede all'utente il **nome del file e il numero della linea**, invia i dati al server e riceve il nuovo contenuto del file, inserendolo nel file system, e stampandolo a video

Il **Server** gestisce in modo parallelo la funzionalità di eliminazione della linea. Per ogni richiesta il processo figlio riceve **il file**, effettua **l'eliminazione** richiesta e **spedisce il risultato** al client.
Ci possono essere problemi di contesa sul file?

NOTA BENE: il server **NON** deve salvare il file in locale, ma agisce direttamente sul flusso di input

PROPOSTA DI ESTENSIONE: SOCKET CONNESSE PER MGET E MPUT

Si vuole abilitare il trasferimento di **una cartella dal server al client** (multiple get - **mget**) e **dal client al server** (multiple put - **mput**) utilizzando la **stessa unica connessione per ogni sessione cliente**

Intendiamo in particolare che si possano trasferire, usando una unica connessione per l'intera sessione di operazione, **intere cartelle in modo piatto, o non ricorsivamente (cioè non entrando nelle sotto-cartelle)**. I direttori inclusi non vengono trattati

Si vuole realizzare la funzionalità spostando i file da un cliente ad un servitore **usando il path corrente** in cui i due processi sono stati invocati

PROPOSTA DI ESTENSIONE: CLIENT

Il **Client**, dopo essersi connesso al server, chiede ciclicamente all'utente il **nome del direttorio**, e se la richiesta sia mget o mput

Nel primo caso di **mget**, il cliente **riceve i file selezionati** (*sia nome sia contenuto*) o un'eventuale risposta negativa, se la cartella non esista lato server. I file richiesti vengono salvati nel path corrente sul processo client, sovrascrivendo file esistenti che abbiano lo stesso nome

Nel secondo caso di **mput**, il cliente **invia i file** (*sia nome sia contenuto*) al server che li deve memorizzare sul proprio file system

Si ricordi che si deve **utilizzare la stessa socket** per il trasferimento di tutti i file. Il cliente chiude la unica connessione solo al termine della sessine, specificata dall'utente con la fine dell'input (filtro)

PROPOSTA DI ESTENSIONE: SERVER

Il **Server** attende una richiesta di connessione da parte dei clienti, poi deve processare le richieste di mget o mput comandate dal cliente. Il servizio deve essere realizzato come **server concorrente** e **parallelo**

Si ricordi che si deve utilizzare la stessa socket per il trasferimento di tutti i file

Il server riceve la richiesta della operazione.
Se è una **mget**, allora riceve il nome della cartella richiesta, e invia, se disponibili, **i file richiesti** o un'eventuale **risposta negativa**

Se è una **mput**, allora riceve i **file** (*nomi e contenuto*) e li scrive sul suo file system. I file richiesti vengono salvati nel direttorio corrente sul server sovrascrivendo file esistenti che abbiano lo stesso nome